Day 3 - Sensors

Overview

In this session, we will learn about different types of sensors and how to interface them with the ESP8266 development board.

There are majorly 3 types of sensors: Analog, Digital with Binary output, and Digital with advanced features. We will understand the difference between these sensors and how we can use them in different IoT applications.

Session flow

Analog sensors

In this part, we will understand the general definition of sensors and how they are helpful for IoT applications. We will also understand how Analog sensors work and what will be required if we want to use one.

Digital sensors with binary output

In this part, we will understand how we can convert an analog sensor into a binary output digital sensor. A lot of applications do not require a controller, instead, we can connect an actuator to a binary output digital sensor.

Digital sensors with advanced features

With the advent of technology, the sensors have become a lot smarter. Nowadays many sensors have advanced features (like; ADC, auto-calibration, temperature compensation, etc) built in. This makes it easy to integrate with the controller. However, these sensors give out data on a certain protocol (Like; I2C, UART, SPI, One-WIRE, etc) and we need to have an understanding of these protocols before we can use these sensors.

Hands-on with different types of sensors using Wokwi

Interface one of the sensors with the actual hardware

Resources

1. Common hardware protocols
2. DS18B20 datasheet
3. DHT11 datasheet

4. MPU6050 datasheet

Daily Challenge from Day 2

Problem Statement:

Write a program to interface RGB LED with ESP8266 and generate different colors.

Hints:

Every pin on RGB LED will require a different GPI
There is a function with the name “value” in the machine module that can be used for changing the intensity of individual colors

Prized Challenge 1

Problem statement:

Build a device to control the user interface of a Windows system using different sensors with ESP8266

Constraints:

- The device should communicate wirelessly with the computer
- You can use any sensor of your choice. However, I recommend using 2 Ultrasonic sensors to implement various gestures.
- Use Python for firmware as well as the software for controlling features on Windows

Resources required:

- Dev board and sensors
- Python modules:
    - PyAutoGUI
    - Networking modules: urequests, pahomqtt, etc.